

# Package ‘hutan’

June 3, 2015

**Title** A Collection of Tools for Phylogenetic Tree Manipulation

**Version** 0.0.0.9000

**Description** A collection of tools for phylogenetic tree manipulation. It is named after the Indonesian word for forest.

**Depends** R (>= 3.1.0)

**License** MIT + file LICENSE

**LazyData** true

**Collate** 'utility\_functions.R'

'hutan.R'

'siphonophore\_constraint.R'

'siphonophore\_ml.R'

**Imports** ape (>= 3.3)

**Suggests** testthat,

roxygen2,

devtools

## R topics documented:

are_bipartitions_compatible . . . . .	2
bipartition_for_edge . . . . .	2
bipartition_for_edge_by_label . . . . .	3
compatible_edges . . . . .	3
descendants . . . . .	4
get_bipartitions . . . . .	4
hutan . . . . .	4
is_compatible_with_set . . . . .	5
is_monophyletic . . . . .	5
siphonophore_constraint . . . . .	6
siphonophore_ml . . . . .	6
tips . . . . .	7
tip_descendants . . . . .	7
zero_constrained . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

`are_bipartitions_compatible`

*Check if two bipartitions drawn from trees with the same tips are compatible with eachother. Each bipartition is defined as a vector of the names of the tips on one side of the bipartition.*

---

### Description

Check if two bipartitions drawn from trees with the same tips are compatible with eachother. Each bipartition is defined as a vector of the names of the tips on one side of the bipartition.

### Usage

```
are_bipartitions_compatible(bi1, bi2, phy)
```

### Arguments

<code>bi1</code>	The first bipartition.
<code>bi2</code>	The second bipartition.
<code>phy</code>	A phylo object describing a tree that includes all tips under investigation. This is used to infer the other half of each bipartition.

### Value

TRUE if `bi1` is compatible with `bi2`, otherwise FALSE.

---

<code>bipartition_for_edge</code>	<i>Get a bipartition, described as a vector of tip numbers, from a specified tree and edge number.</i>
-----------------------------------	--

---

### Description

Get a bipartition, described as a vector of tip numbers, from a specified tree and edge number.

### Usage

```
bipartition_for_edge(phy, edge)
```

### Arguments

<code>phy</code>	A phylo object that specifies the tree.
<code>edge</code>	The number of the edge that defines the bipartition.

### Value

A vector of tip nodes (specified by numbers) that define one half of the bipartition (the other half is the set of tip nodes that are not in this vector).

---

bipartition\_for\_edge\_by\_label

*Get a bipartition, described as a vector of tip labels, from a specified tree and edge number.*

---

### Description

Get a bipartition, described as a vector of tip labels, from a specified tree and edge number.

### Usage

```
bipartition_for_edge_by_label(edge, phy)
```

### Arguments

edge	The number of the edge that defines the bipartition.
phy	A phylo object that specifies the tree.

### Value

A vector of tip nodes (specified by labels) that define one half of the bipartition (the other half is the set of tip nodes that are not in this vector).

---

compatible_edges	<i>Identify the edges in one phylo object that are compatible with the edges in another phylo object. Requires the same tip labels for each tree.</i>
------------------	---

---

### Description

Identify the edges in one phylo object that are compatible with the edges in another phylo object. Requires the same tip labels for each tree.

### Usage

```
compatible_edges(phy1, phy2)
```

### Arguments

phy1	The tree under consideration
phy2	The tree to be compared to

### Value

A boolean vector corresponding to the edges in phy1. Each element is FALSE if the edge is incompatible with phy2, or TRUE if compatible.

---

descendants	<i>Get all the descendants of a given node in a tree.</i>
-------------	---

---

### Description

Get all the descendants of a given node in a tree.

### Usage

```
descendants(phy, a, keep_node = FALSE)
```

### Arguments

phy	A phylo object that specifies the tree.
a	The number of a node in phy.
keep_node	If FALSE, do not include a in the result.

### Value

A vector of nodes (specified by number) that are descendants of a. Includes internal and tip nodes.

---

get_bipartitions	<i>Get a list of all the bipartitions in a tree.</i>
------------------	--

---

### Description

Get a list of all the bipartitions in a tree.

### Usage

```
get_bipartitions(phy)
```

### Arguments

phy	A phylo object that specifies the tree.
-----	---

### Value

A list of bipartitions for the tree. The order of the list corresponds to the edges in phy\$edge. Bipartitions are specified as a vector of the tip labels that make up one half of the bipartition.

---

hutan	<i>hutan: A collection of tools for phylogenetic tree manipulation.</i>
-------	---

---

### Description

The hutan package provides functions for common phylogenetic tree manipulation tasks, and uses these facilitate some more specialized tasks. It is named after the Indonesian word for forest.

---

is\_compatible\_with\_set

*Check if bipartition bi is compatible with the bipartitions in bi\_list. Each bipartition is defined as a vector of the names of the tips on one side of the bipartition.*

---

### Description

Check if bipartition bi is compatible with the bipartitions in bi\_list. Each bipartition is defined as a vector of the names of the tips on one side of the bipartition.

### Usage

```
is_compatible_with_set(bi, bi_list, phy)
```

### Arguments

bi	The query bipartition.
bi_list	A list of the bipartitions to be compared against.
phy	A phylo object describing a tree that includes all tips under investigation. This is used to infer the other half of each bipartition.

### Value

TRUE if bi is compatible with all bipartition in bi\_list, otherwise FALSE.

---

is\_monophyletic

*Test if a set of tips, specified as a vector of tip labels, forms a monophyletic group in a given tree. The test is unrooted, i.e. the group can span the root.*

---

### Description

Test if a set of tips, specified as a vector of tip labels, forms a monophyletic group in a given tree. The test is unrooted, i.e. the group can span the root.

### Usage

```
is_monophyletic(phy, x)
```

### Arguments

phy	The tree under consideration
x	A vector of the labels of the tips in question

### Value

A boolean, TRUE if the tips form a monophyletic group.

---

siphonophore\_constraint

*Siphonophores constraint phylogeny.*

---

**Description**

An unresolved phylogeny that constrains the group Agalmatidae sensu stricto + Bargmannia to be monophyletic, corresponding to the published SOWH test

**Usage**

siphonophore\_constraint

**Format**

An ape phylo object

**Source**

<http://dx.doi.org/10.1080/10635150500354837>

---

siphonophore\_ml

*Siphonophores phylogeny.*

---

**Description**

A maximum likelihood phylogeny of siphonophores

**Usage**

siphonophore\_ml

**Format**

An ape phylo object

**Source**

<http://dx.doi.org/10.1080/10635150500354837>

---

tips	<i>Get tips and labels of a phylo object.</i>
------	---

---

**Description**

Get tips and labels of a phylo object.

**Usage**

```
tips(phy)
```

**Arguments**

phy	A phylo object.
-----	-----------------

**Value**

A vector of all the tips, annotated with their names

---

tip_descendants	<i>Get all the tips that are descendants of a given node in a tree.</i>
-----------------	---

---

**Description**

Get all the tips that are descendants of a given node in a tree.

**Usage**

```
tip_descendants(phy, a)
```

**Arguments**

phy	A phylo object that specifies the tree.
a	The number of a node in phy.

**Value**

A vector of tip nodes (specified by number) that are descendants of a. If a is a tip, it is the sole element of this vector.

---

zero_constrained	<i>Generates the "zero-constrained" tree described by Susko 2014 (<a href="http://dx.doi.org/10.1093/molbev/msu039">http://dx.doi.org/10.1093/molbev/msu039</a>)</i>
------------------	--

---

**Description**

Generates the "zero-constrained" tree described by Susko 2014 (<http://dx.doi.org/10.1093/molbev/msu039>)

**Usage**

```
zero_constrained(phy_resolved, phy_constraint, epsilon = 1e-06)
```

**Arguments**

`phy_resolved` A fully resolved phylogeny stored as a phylo object, e.g. an ML tree.  
`phy_constraint` A partially resolved constraint tree.  
`epsilon` The value to replace the branch length with

**Value**

A phylo object containing a tree that is the same as `phy_resolved`, except that the length of edges that are incompatible with `phy_constraint` are replaced with `epsilon`.

**Examples**

```
data( siphonophore_ml )  
data( siphonophore_constraint )  
zc <- zero_constrained( siphonophore_ml, siphonophore_constraint )
```



# Index

## \*Topic **datasets**

siphonophore\_constraint, [6](#)

siphonophore\_ml, [6](#)

are\_bipartitions\_compatible, [2](#)

bipartition\_for\_edge, [2](#)

bipartition\_for\_edge\_by\_label, [3](#)

compatible\_edges, [3](#)

descendants, [4](#)

get\_bipartitions, [4](#)

hutan, [4](#)

hutan-package (hutan), [4](#)

is\_compatible\_with\_set, [5](#)

is\_monophyletic, [5](#)

siphonophore\_constraint, [6](#)

siphonophore\_ml, [6](#)

tip\_descendants, [7](#)

tips, [7](#)

zero\_constrained, [8](#)